STIC Search Report
EIC 2100

USPTO

STIC Database Tracking Number: 107232

TO:	From:
TO: James Seal **Location: 4D11** **Art Unit : 2131** **Friday, October 31, 2003** **Case Serial Number: 09/827386**	**From: David Holloway** **Location: EIC 2100** **PK2-4B30** **Phone: 308-7794** **david.holloway@uspto.gov**

Search Notes

Dear Examiner Seal,

Attached please find your search results for above-referenced case.
Please contact me if you have any questions or would like a re-focused search.

David

sira
Search and Information Resources Administration

STIC EIC 2100 Search Request Form

107232
147

Today's Date: 31 Oct 2003

What date would you like to use to limit the search?
Priority Date: Aug 1 96 Other:

Name James Seal

AU 2131 Examiner # 76900

Room # 4D11 Phone 308-4562

Serial # 09827386

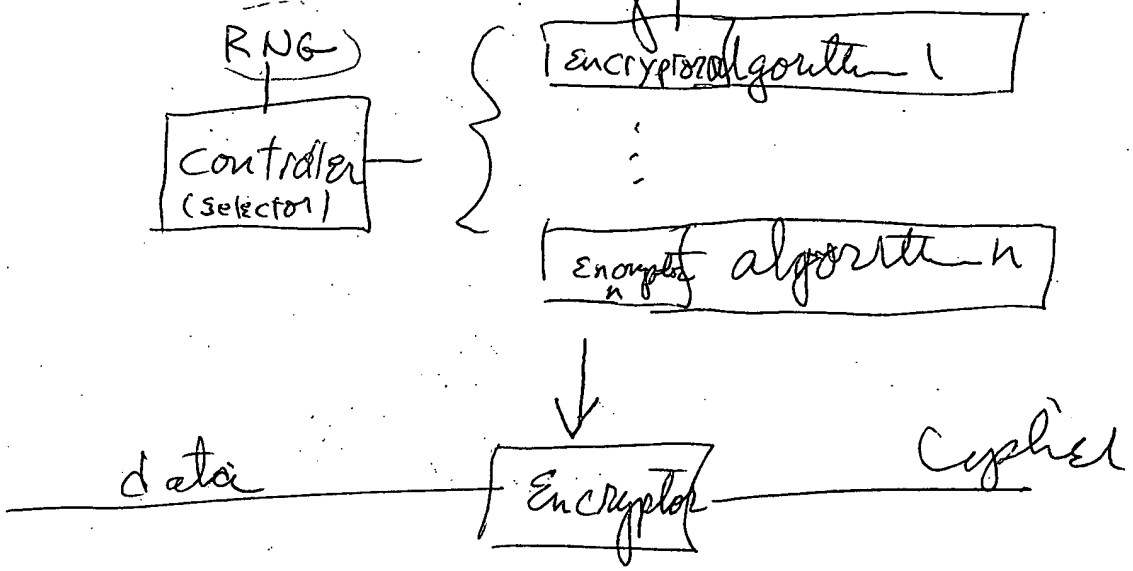
Format for Search Results (Circle One):
PAPER DISK EMAIL

Where have you searched so far?
USP DWPI EPO JPO ACM IBM TDB
IEEE INSPEC SPI Other

Is this a "Fast & Focused" Search Request? (Circle One) YES NO
A "Fast & Focused" Search is completed in 2-3 hours (maximum). The search must be on a very specific topic and meet certain criteria. The criteria are posted in EIC2100 and on the EIC2100 NPL Web Page at http://ptoweb/patents/stic/stic-tc2100.htm.

What is the topic, novelty, motivation, utility, or other specific details defining the desired focus of this search? Please include the concepts, synonyms, keywords, acronyms, definitions, strategies, and anything else that helps to describe the topic. Please attach a copy of the abstract, background, brief summary, pertinent claims and any citations of relevant art you have found.

Need selectable encryption



STIC Searcher David A. Illoway Phone 308-7794

Date picked up 10-31-03 Date Completed 10-31-03

```
Set     Items   Description
S1      26488   (ENCRYPT? OR ENCIPHER? OR CIPHER? OR CYPHER? ? OR ENCYPHER-
             ?)
S2       5460   S1(3N)(ALGORITHM? OR SYSTEM? OR FUNCTION? OR SCRAMBL? OR M-
             ATHEMATIC?()RULE?)
S3        238   (MULTIPL? OR PLURAL? OR SEVERAL? OR MANY OR VARIOUS OR VAR-
             IET?)(5N)S2
S4    1114371   RANDOM? OR PSUEDORANDOM? OR RNG? ?
S5         76   SIDE(N)SIDE
S6          7   (TELEPHON? OR PHONE?)()SCRAMBL?
S7    4716075   SELECT? OR ROTATE? OR CYCLE? OR SWITCH? OR ALTERNATING OR -
             CHOOSE? OR DESIGNAT?
S8         63   S3 AND (S4 OR S7)
S9        162   (TELEPHON? OR TELECOM? OR PHON? OR MODEM)(5N)(S5 OR SCRAMB-
             L?)
S10         0   S5 AND S2
S11         0   S1 AND S5
S12       137   RD S9 (unique items)
S13       100   S12 NOT PY>1996
S14        98   S13 NOT PD=19960801:19990601
S15        98   S14 NOT PD=19990601:20031101
S16        13   S15 AND S1
S17        41   RD S8 (unique items)
S18        12   S17 NOT PY>1996
S19        11   S18 NOT PD=19960801:19990601
S20        11   S19 NOT PD=19990601:20031101
File   8:Ei Compendex(R) 1970-2003/Oct W3
         (c) 2003 Elsevier Eng.  Info. Inc.
File  35:Dissertation Abs Online 1861-2003/Sep
         (c) 2003 ProQuest Info&Learning
File  65:Inside Conferences 1993-2003/Oct W4
         (c) 2003 BLDSC all rts. reserv.
File   2:INSPEC 1969-2003/Oct W3
         (c) 2003 Institution of Electrical Engineers
File  94:JICST-EPlus 1985-2003/Nov W1
         (c)2003 Japan Science and Tech Corp(JST)
File 111:TGG Natl.Newspaper Index(SM) 1979-2003/Oct 28
         (c) 2003 The Gale Group
File 233:Internet & Personal Comp. Abs. 1981-2003/Jul
         (c) 2003, EBSCO Pub.
File 144:Pascal 1973-2003/Oct W3
         (c) 2003 INIST/CNRS
File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
         (c) 1998 Inst for Sci Info
File  34:SciSearch(R) Cited Ref Sci 1990-2003/Oct W4
         (c) 2003 Inst for Sci Info
File  62:SPIN(R) 1975-2003/Sep W2
         (c) 2003 American Institute of Physics
File  99:Wilson Appl. Sci & Tech Abs 1983-2003/Sep
         (c) 2003 The HW Wilson Co.
```

20/5/9 (Item 5 from file: 2)
DIALOG(R)File 2:INSPEC

02223901 INSPEC Abstract Number: B84020984, C84015494
Title: Randomized **encryption techniques**
Author(s): Rivest, R.L.; Sherman, A.T.
Author Affiliation: Lab. for Computer Sci., MIT, Cambridge, MA, USA
Conference Title: Advances in Cryptology, Proceedings of Crypto 82 p. 145-63
Editor(s): Chaum, D.; Rivest, R.L.; Sherman, A.T.
Publisher: Plenum, New York, NY, USA
Publication Date: 1983 Country of Publication: USA xv+331 pp.
ISBN: 0 306 41366 3
Conference Date: 23-25 Aug. 1982 Conference Location: Santa Barbara, CA, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Theoretical (T)
Abstract: A **randomized** encryption procedure enciphers a message by **randomly** choosing a ciphertext from a set of ciphertexts corresponding to the message under the current encryption key. At the cost of increasing the required bandwidth, such procedures may achieve greater cryptographic security than their deterministic counterparts by increasing the apparent size of the message space, eliminating the threat of chosen plaintext attacks, and improving the a priori statistics for the inputs to the **encryption algorithms** . This paper explores **various** ways of using **randomization** in encryption. (30 Refs)
Subfile: B C
Descriptors: cryptography
Identifiers: **randomized** encryption procedure; ciphertext; cryptographic security
Class Codes: B6120B (Codes); C1260 (Information theory)

```
Set     Items   Description
S1      13828   (ENCRYPT? OR ENCIPHER? OR CIPHER? OR CYPHER? ? OR ENCYPHER-
             ?)
S2       3004   S1(3N)(ALGORITHM? OR SYSTEM? OR SCRAMBL? OR MATHEMATIC?()R-
            ULE?)
S3         86   (MULTIPL? OR PLURAL? OR SEVERAL? OR MANY OR VARIOUS OR VAR-
              IET?)(5N)S2
S4      75839   RANDOM? OR PSUEDORANDOM? OR RNG? ?
S5        271   SIDE(N)SIDE
S6          5   (TELEPHON? OR PHONE?)()SCRAMBL?
S7    1430165   SELECT? OR ROTATE? OR CYCLE? OR CHOOSE? OR DESIGNAT?
S8         29   S3 AND (S4 OR S7)
S9         21   S8 NOT AD=19960801:19990601
S10        21   S9 NOT AD=19990601:20000601
S11        13   S10 NOT AD=20000601:20031101
S12        89   (TELEPHON? OR TELECOM? OR PHON? OR MODEM)(5N)(S5 OR SCRAMB-
              L?)
S13        79   S12 NOT AD=19960801:19990601
S14        65   S13 NOT AD=19990601:20031101
S15         0   S14 AND S3
S16         6   S14 AND S1
S17       159   S1(N)FUNCTION?
S18         2   S17 AND S3
S19         3   S3 AND SWITCH?
S20         1   S19 NOT S11
File 350:Derwent WPIX 1963-2003/UD,UM &UP=200369
         (c) 2003  Thomson Derwent
```

11/5/1
DIALOG(R)File 350:Derwent WPIX

011769109 **Image available**
WPI Acc No: 1998-186019/199817
XRPX Acc No: N98-147862

Data encoding and decoding method for computer - involves decoding enciphered sentence through use of decoding program and key data after separating decoding program and enciphered sentence received by data receiver via network or recording medium

Patent Assignee: OKAMOTO E (OKAM-I); TOSHIBA KK (TOKE)
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 10041934	A	19980213	JP 96197854	A	19960726	199817	B

Priority Applications (No Type Date): JP 96197854 A 19960726
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 10041934	A		7	H04L-009/14	

Abstract (Basic): JP 10041934 A

The method involves **selecting** one of **several encrypted algorithms** which correspond to received data. Sentence data (M) are enciphered by using key data (Ks), in order to obtain enciphered sentences (C). A decoding program, which corresponds to the **selected** algorithm, is synthesised with the enciphered sentence.

The combined decoding program and enciphered sentence are received and separated by a data receiver via a network or a recording medium. The enciphered sentence is decoded through the use the decoding program and key data, in order to obtain the sentence data.

ADVANTAGE - Enables easy encoding and decoding operation corresponding to different types of data.

Dwg.4/4

Title Terms: DATA; ENCODE; DECODE; METHOD; COMPUTER; DECODE; ENCIPHER; SENTENCE; THROUGH; DECODE; PROGRAM; KEY; DATA; AFTER; SEPARATE; DECODE; PROGRAM; ENCIPHER; SENTENCE; RECEIVE; DATA; RECEIVE; NETWORK; RECORD; MEDIUM
Derwent Class: P85; T01; W01
International Patent Class (Main): H04L-009/14
International Patent Class (Additional): G06F-013/00; G09C-001/00; H04L-009/08; H04L-009/36
File Segment: EPI; EngPI

11/5/6
DIALOG(R)File 350:Derwent WPIX

010445093 **Image available**
WPI Acc No: 1995-346410/199545
XRPX Acc No: N95-259030
Data encryption method - encrypting data into number of data blocks and control blocks which define data block encryption using randomly selected **functions**
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC); IBM CORP (IBMC)
Inventor: YORKE-SMITH I E
Number of Countries: 005 Number of Patents: 005
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 676876	A1	19951011	EP 95302218	A	19950403	199545	B
GB 2288519	A	19951018	GB 946613	A	19940405	199545	
JP 7281596	A	19951027	JP 9533219	A	19950222	199601	
US 5548648	A	19960820	US 94276192	A	19940715	199639	N
JP 3229148	B2	20011112	JP 9533219	A	19950222	200174	

Priority Applications (No Type Date): GB 946613 A 19940405; US 94276192 A 19940715
Cited Patents: 02Jnl.Ref; DE 1447301; EP 464562; EP 95923; JP 5102960; US 5253294
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 676876	A1	E	16	H04L-009/00	
Designated States (Regional): DE FR GB					
GB 2288519	A		24	H04L-009/14	
JP 7281596	A		12	G09C-001/00	
US 5548648	A		13	H04L-009/00	
JP 3229148	B2		11	G09C-001/00	Previous Publ. patent JP 7281596

Abstract (Basic): EP 676876 A

The encryption method for encrypting data containing data segments (DS1-DSn) into encrypted data blocks (EDB1-EDBn) and associated control blocks (CB1-CBn) involves **selecting** several encryption functions (F1-Fi). The data segments are encrypted using the **selected** function. The data segments can be of varying lengths. An encrypted data block is produced containing the encrypted data segment. An associated control block is produced for each encrypted data block. Each control block consists of the information needed to decrypt the data blocks.

Pref., the encryption functions are chosen via a mapping of **random** numbers to functions. The data, its start point in the block and its length are all encrypted. Both blocks are padded with **random** numbers and start positions can vary.

USE/ADVANTAGE - Communication systems. Provides encryption system that is computationally efficient while retaining high security. Short encryption and decryption times.

Dwg.3/7

Title Terms: DATA; ENCRYPTION; METHOD; DATA; NUMBER; DATA; BLOCK; CONTROL; BLOCK; DEFINE; DATA; BLOCK; ENCRYPTION; **RANDOM** ; **SELECT** ; FUNCTION
Derwent Class: P85; W01
International Patent Class (Main): G09C-001/00; H04L-009/00; H04L-009/14
International Patent Class (Additional): H04L-009/06; H04L-009/12; H04L-009/16
File Segment: EPI; EngPI

11/5/11
DIALOG(R)File 350:Derwent WPIX

008332139 **Image available**
WPI Acc No: 1990-219140/199029
XRPX Acc No: N90-170043

Certification system for IC card memory - sends random **number, encryption algorithm** selector **and key data between terminal and card to certify terminal**

Patent Assignee: TOSHIBA KK (TOKE)
Inventor: IIJIMA Y
Number of Countries: 004 Number of Patents: 005
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
GB 2227111	A	19900718	GB 8929239	A	19891228	199029	B
JP 2187785	A	19900723	JP 898011	A	19890117	199035	
FR 2641885	A	19900720				199036	
GB 2227111	B	19930519	GB 8929239	A	19891228	199320	
US 5293029	A	19940308	US 90463601	A	19900111	199410	
			US 91747420	A	19910819		
			US 92942337	A	19920909		

Priority Applications (No Type Date): JP 898011 A 19890117; JP 898010 A 19890117
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5293029	A		22	G06K-005/00	Cont of application US 90463601
					Cont of application US 91747420
GB 2227111	B			G07F-007/08	

Abstract (Basic): GB 2227111 A

The certification system includes an electronic device with at least one key data. A second electronic device is capable of performing communication with the first electronic device. The first data and **designation** data fro **designating** key data for encrypting the first data is transmitted from the second electronic device to the first electronic device.

When the first data and the **designation** data are received by the first electronic device, one key data from the at least one key data in accordance with the received **designation** data is **selected** and the received first data is encrypted by using the **selected** key data. Part of the encrypted data is transmitted to the second electronic device after the first data is entirely received by the first electronic device.

USE - For IC cards using erasable non-volatile and control element.

Dwg.1/8

Title Terms: CERTIFY; SYSTEM; IC; CARD; MEMORY; SEND; **RANDOM** ; NUMBER; ENCRYPTION; ALGORITHM; **SELECT** ; KEY; DATA; TERMINAL; CARD; CERTIFY; TERMINAL
Derwent Class: P85; T01; T04; T05
International Patent Class (Main): G06K-005/00; G07F-007/08
International Patent Class (Additional): G06K-019/07; G09C-001/00; H04L-009/14; H04L-009/32
File Segment: EPI; EngPI

5/9/3 (Item 3 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01103770 SUPPLIER NUMBER: 00588746 (THIS IS THE FULL TEXT)
The Prolok **Plus Scheme Has The Potential To Be Hard On Innocent Users.**
Norton, P.
PC Week, v1, n45, p75
Nov. 13, 1984
DOCUMENT TYPE: column ISSN: 0740-1604 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 994 LINE COUNT: 00072

ABSTRACT: **Prolok** Plus is security software which uses copy protection techniques. It has the ability to detect an illegal copy whose code has been tampered with in an attempt to make the illegal copy runnable. **Prolok** Plus will not run such a copy, but also instructs users to remove the illegal disk within ten seconds or the entire disk will be destroyed. **Prolok** is exposing a legal user to enormous grief and itself to problems of product liability. This is supposed to deter software piracy, but it does more damage to the legal users.

TEXT:

As we discussed last week, **Prolok** Plus adds to the **Prolok** copy-protection scheme the ability to detect an illegal copy whose code has been tampered with in an attempt to make the illegal copy runnable.

Prolok Plus quite naturally refuses to run such a copy.

It also instructs you to remove the illegal disk within 10 seconds, or "OTHER PORTIONS OF THE DISK WILL BE **RANDOMLY** DESTROYED."

Prolok Plus, you see, is now ready to retaliate by planting on the disk a "worm" that will punish the guilty by **randomly** destroying data.

Twenty-megabyte hard disk or 400K-byte floppy, this message is telling you **Prolok** Plus will eat them up if you don't get that disk out of there and fast. war on Copies?

Did I miss the "Manufacturer's Declaration of War on Users"? Should I buy a gun?

Most people are already afraid of computers. The major thrust in software development for personal computers has been to give them a "friendly face." That effort has been undertaken with the hope of dissolving the fear that computer innocents have of this unknown technology.

To that end, there are notices in IBM's manuals to the effect that nothing you can do will hurt the machine and that you should experiment. We have TopView; we have Lisa, Macintosh and mice; we have integrated software and easy-to-learn interfaces.

Now, we also have a program that says it's going to destroy your disk if you don't act fast. Who Is It Really Hurting?

Serves the pirates right, you say? Maybe it would serve the pirates right, but they're the one group it won't hurt at all-- **Prolok** 's machinations might actually help them, as we shall see later.

Even if it could hurt them, I'd be happier if the punishment were meted out after a trials, a nicety for which most judicial bodies see a need.

Who's to say that the holder of that "illegal" copy is not a (possibly new) user who legally bought a **Prolok** Plus-protected disk, legally bought a backup program, legally copied the **Prolok** disk to a hard disk to be able to use it more conveniently--a reasonable thing to want to do, wouldn't you say?

Remember that it is legal to make backup copies for your own use and to buy programs that help you make backup copies.

What happens when these customers, out several hundred dollars for a new program, see the aforementioned message that their disks will be damaged if they don't remove the offending illegally copied disk quickly?

And how, by the way, should they "remove" their hard disks? Would you or I know what to do?

Suppose, as is likely, that the disk doesn't get removed before the end of **Prolok** 's countdown, either because the user wasn't watching the screen, moved too slowly, or couldn't figure out how to "remove" a hard

disk, and data really is destroyed? Who suffers then?

I think **Prolok** is exposing a legal user to enormous grief and itself to enormous problems of product liability. The concept of the merciless, punitive machine may have appeared in 1984, but that was just a book, for goodness sake; personal computers were said to be proving how wrong parts of that book were, not how prescient they were. What IS the Purpose?

the part that bothers me most is that the destruction of data serves no purpose, other than simple revenge.

Prolok has already served its purpose by detecting the illegal copy; it could simply destroy the rest of the program and thus entirely wipe away any chance of using the illegal copy.

Clearly, since **Prolok** is not taking the logical step simply destroying the illegally copied program, the company must be hoping that the psychological impact of this message--which will spread far and wide by word of mouth--will deter pirates.

Here, again, **Prolok** 's thinking seems to be severely muddied: A real pirate will know what he's fooling with.

A real pirate will be running the illegal copy off a floppy with nothing else on it and will look upon this message as a challenge. He'll probably flip his machine into "single step" mode and watch in amusement as this now tamed device starts off on its doomed machinations.

In a day or two or three, any pirate worth his sea legs will have a patch to change the code that checks to see if the code was changed.

The way to slow down the pirate would be to destroy the illegal program immediately and with no warning. Are They Bluffing?

It becomes reasonable to wonder whether maybe this whole nasty thing isn't just **Prolok** 's bluff, since that would accomplish the psychological goal without **Prolok** risking any liability.

I don't know, but I hope it's not. Software companies--or any companies, for that matter--have no business trying to accomplish their ends by scaring their customers or, worse, willfully damaging customers' property. I know of no company, in this country at least, that finds it necessary to scare and hurt its customers in order to stay in business. I certainly see no reason why a software company should need to do so.

We have all had various locations in our disks go bad. What would happen if a byte in the routine that **Prolok** Plus checks just happened to go bad?

We have enough troubles with accidental bugs. I just can't imagine anyone deliberately adding something to their systems that says: "If you misuse me, I will willfully cause damage."

It's impossible to tell if a program you have is protected by **Prolok** Plus until you raise its wrath, but I expect that word will get around.

DESCRIPTORS: Systems/Data Security Software; Copy Prevention Techniques; Product Liability; Software Piracy
TRADE NAMES: **Prolok** Plus
FILE SEGMENT: CD File 275

5/9/1 (Item 1 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01176444 SUPPLIER NUMBER: 00656168 (THIS IS THE FULL TEXT)
The Copy-Protection Wars.
Taylor, Jared.
PC Magazine, v5, n1, p165-167
Jan. 14, 1986
ISSN: 0888-8507 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 5111 LINE COUNT: 00376

ABSTRACT: A computer program can be protected by software which establishes a specific magnetic pattern on the original software, and provides a check program to read this pattern. Software-supported protection plans have been copied successfully by copying programs, which use hardware, such as add-on boards, to solve the problem of software protection schemes. The hardware protection schemes used by companies are to burn tiny holes in the disk with a laser, use a ROM chip with a special program that acts as a key, and a special device that, when held up to the screen, unscrambles the program. There are methods available to users to protect data with programs that alter the directory table so it cannot be read by DOS, a computer data encryption algorithm, and permutation algorithms. The problems with software encryption can be solved by encryption on a chip, which enables faster access than software encryption methods.

TEXT:

THE COPY-PROTECTION WARS

Quietly but constantly, the war rages between those who protect software and those who copy it. It's a fascinating struggle of brains and imagination, with each side in business to outwit the other. Like a miniature arms race, each new weapon quickly gives rise to another, and countermeasures are foiled with counter-countermeasures. Lately there's been escalation as both sides roll out hardware in a fight that until now has mostly been fought with a software arsenal.

The struggle for data security rages on the same battlefield. As personal computer use spreads, more and more users need to protect sensitive data. Here, the first round has clearly gone to the protectors. Even inexpensive data-protection programs use exotic encryption methods that may be foolproof. In just a few seconds, you can scramble a file so thoroughly that not even the C.I.A. can read it. However, the real threat to your data can come from unexpected sources.

The problem of unauthorized copying is especially serious for the microcomputer industry for two reasons. First, software and data can be extremely valuable. Second, anyone with a computer can make perfect copies of originals. Record companies would have the same problem if records cost hundreds of dollars and anyone with a record player could make a perfect copy for next to nothing.

Software copying is such a temptation that, if we're to believe the software industry, nearly everyone is doing it. According to a survey done in January by Future Computing, one pirated copy of business software is in use for every legitimate copy. If you make the conservative assumption that one in four pirate users would have bought the program if unable to copy it, then last year the software industry lost $600 million to piracy. No wonder vendors protect their programs.

Two Kinds of Protection

There are two ways to protect a computer program--with hardware or software. Classic software-based protection itself has two parts. The first element is a magnetic pattern or "fingerprint' in one spot on each original copy of a program disk. The second part is a small check program that looks for this fingerprint. When you try to start the protected program, the check routine runs first. If it finds the fingerprint, it figures the disk is an original and loads the application program.

What happens when you use DOS to copy a protected disk? Everything will copy perfectly except for the fingerprint. When you try to run the program, the check routine won't find the fingerprint and therefore won't run the application program. The copy protection is thus based on the fact

that the NEC disk controller used in the IBM PC can recognize the odd magnetic pattern of a fingerprint on the original disk but can't reproduce it on a copy.

How is it that a system that normally makes perfect copies can read things it can't write? Rick Landuyt, president of a copy-protection company called Glenco Engineering, explains: "Electronically speaking, reading and writing are entirely different. They're separate circuits, just as your eyes and ears are separate circuits. You can see the color red, but you can't hear it. In the same way, the PC reads things it can't write.' In effect, the fingerprint on a protected disk is a disk error. It's useful for DOS to be able to detect errors, but there's no reason for it to replicate them.

Companies that make protection schemes have invented many different kinds of fingerprints. The IBM PC formats a blank disk so that it writes data in separate tracks. Each track is divided into eight or nine sectors (depending on the version of DOS) with 512 bytes per sector. A typical fingerprint might be a track made up of a peculiar number of sectors or a series of sectors that are an odd size. One of the most common schemes is to physically overlay a series of sectors. Another is to give a sector a strange ID.

Another fingerprinting technique uses weak bits. These bits are of unstable data that have a different value every time they're read. In a copy-protection scheme, the program reads the weak bits several times in a row. If the program gets different values for the data, it knows the data is weak and that the disk is an original. A DOS DISKCOPY command doesn't produce weak bits; it writes permanent values in the sector that was supposed to be weak. Thus, if the check routine finds strong bits that return the same value every time they are read, it knows the disk is a copy and will refuse to load.

Some fingerprints are harder to produce than others. A standard PC can actually be made to write simple fingerprints if you use special software to drive the disk controller. Both Glenco Engineering and another well-known copy-protection company, Softguard Systems Inc., sell kits that turn a PC into a copy-protection machine. Strictly through software, they make the disk controller write oddball tracks that DOS normally doesn't allow. Disks with these fingerprints can't be copied using standard DOS utilities.

However, anything that the PC can be made to write with special software, it can be made to copy with special software. Such low-level protection schemes are easily defeated by the many good copying programs that are now available. Disk-Tech, Quaid Software, and Central Point Software all make copy programs that act like the kits sold by the copy protectors (see sidebar, "The Master Copiers'). They analyze the original disk's fingerprint and tell the disk controller to copy it. What software can do, other software can undo.

The copy-protection companies have fought back by using special machines that lay down tracks that the PC's disk controller can read but can't possibly be made to copy. This is, of course, more of a bother for the software producer because it can't duplicate this protection system itself. It must either buy specially formatted disks from the protection company or get the protection company to do the duplicating.

Even so, protection may not be perfect, since the copy-busting program may not have to make a perfect copy of the fingerprint. Kevin Larsen, sales manager of Disk-Tech, which produces Personal Copier, gives the example of a copy-protection scheme that might write 101 sectors to the fingerprint track instead of the usual 8 or 9. Even with special software, the NEC controller can't write a track with 101 sectors. However, as Larsen explains, when the protection routine checks the fingerprint, all it can actually look for is the address of the 101st sector rather than count all 101 to be sure they're there.

To defeat such a scheme, all you have to do is write an address for sector 101 rather than format 101 physical sectors. This the NEC controller card can do. The fingerprint check will then find what it's looking for and be fooled into loading the program.

Getting Tough

It didn't take the protection companies long to get wise to this trick. Now they make programs that run a thorough check for a fingerprint that the PC itself can't possibly write. But still, the copiers can best

them. Copy Write, by Quaid Software, has a utility that doesn't even try to copy an ornery fingerprint. Instead, it puts a description of the fingerprint on the new disk rather than the real thing. Before you run the copy, you first load a small, memory-resident Quaid utility. At run time, the utility interrupts the protection-checking routine, reads the description of the uncopyable fingerprint from the new disk, and tells the check routine, "Here's what you're looking for.' The protection program thinks it's found the real thing and loads the program.

Yet another software-based protection scheme starts with a scrambled, unusable version of the program on disk. As usual, the protection routine first looks for a fingerprint. Part of the fingerprint is a key that unscrambles the program after it's already in memory. Only then will it run.

Bob McQuaid, president of Quaid Software, explains that even if it isn't possible to copy the fingerprint and key perfectly, there's a way to defeat this scheme. A clever utility can capture the protected program after it's been unscrambled and write it out to disk. The unscrambled program can then be copied with no trouble. "We have not found anything that we cannot copy,' says McQuaid.

However, as software protection schemes proliferate, it gets harder to design a single program to beat them all. The copy busters have neatly solved the problem with hardware. Central Point and Disk-Tech have both introduced add-on boards for the PC that make perfect magnetic copies of any disk. Central Point's board costs $95 and is cabled to both the disk controller and the drive. When Central Point software is running, the board takes over from the controller. Since it has none of the write limitations of the NEC disk controller, it can copy anything. As Central Point president Michael Brown explains, "It just picks up the data from one disk, puts it down on another and says, "Next sector, please.'

Joe Diodati, Softguard's vice president for marketing and sales, takes a dim view of the new boards. "We've been pretty good at being a moving target,' he says, "but this is an escalation is the war.' Diodati says his company is not planning a hardware counterattack but is trying to find a software technique to outwit the copying boards. Even so, he is generous to the competition. "You've got to give Central Point a hell of a lot of credit,' he says. "What they have done is produce a cheap duplication system--it's a nice piece of equipment.'

On the Hard Disk Front

As the floppy disk battle rages, another front has opened up on hard disks. More and more software is designed to run on a hard disk, but that poses a different kind of protection problem. So far, many vendors have taken the key-disk approach. You can copy the protected program onto as many hard disks as you like, but in order to run it, you have to put the original floppy in the A: drive. When the program runs from the hard disk, it checks for the fingerprint just as if it were running from the floppy.

Since some of the most popular programs (1-2-3, Framework, Symphony, dBASE III) use this technique, there is incentive to defeat it. The most common method is to use a small, memory-resident utility that sits and waits for the protection routine to hunt for the disk in drive A: The utility ambushes the query, tells it that all is well in drive A: and sends it home. The check thinks it's done its job and runs the program.

Quaid and Nostradamus Inc. both sell memory-resident programs that defeat the key-disk check for some popular programs. Central Point's Copy II PC includes a similar utility at no extra charge.

Lately, the hard disk battle has taken a new turn, since legitimate users are tired of key disks. The protection companies have responded with a way to install protected programs so that they run without a key disk while remaining protected. Such programs come with an installation utility that counts how many times you put the program on a hard disk. Two or three shots is all you get. However, since you might decide to move the program to a different hard disk, there's usually an "uninstall' routine to remove the program. Each time you uninstall, you get another chance to reinstall onto a different hard disk.

This technique could work well for site licenses. A large company that wants 500 copies of a program could strike a deal with a software company for a custommade floppy disk that allowed 500 hard disk installations instead of the usual 2. The deal could include a fat discount on the cost per copy.

Hard disk installation uses the old fingerprint technique. However, since the fingerprint has to be on each user's hard disk, each PC has got to write it. As I discussed earlier, any fingerprint a PC can be made to write on a floppy it can be made to copy--with special sofware. This sounds like a copy-buster's dream. It's not. Each hard disk fingerprint includes information about the unique physical characteristics of the disk.

As it happens, hard disks are rarely, if ever, identical. Each has a different number of bad tracks in different locations. Thus, if the fingerprint contains this information, even using DISKCOPY to duplicate the contents of one hard disk on another will not make a good copy. The fingerprint on the second disk won't match its physical characteristics and the program won't run.

One way around this, of course, is to make faithful magnetic copies of the program disk before you use it to install to a hard disk. Then you could use up all the installations on the original disk and still have live copies that will also install. The battle of wits goes on.

The Hardware Wars

Physical copy-protection schemes are a different matter. Of these, the best known is certainly **PROLOK** by Vault Corp. **PROLOK** works by burning one or two tiny holes in the protected disk with a laser. This branding makes a physical fingerprint that even the new copy boards can't reproduce. If the check routine doesn't find the little holes, the program won't run.

This hole scheme souns like bulletproof protection, but it's not. Quaid Software took advantage of the fact that **PROLOK** was making a series of checks for the laser burn rather than a single pass. During one of those checks, a Quaid program fooled **PROLOK** into thinking it had found the burn. Vault has taken Quaid Software to court for breaking its protection scheme.

Vault chairman W. Krag Brotby says he knows of no copy program that can defeat the latest version of **PROLOK** . He says the company puts out a new version "roughly every quarter' to take advantage of new releases of DOS and to keep ahead of the copy busters.

Vault also sells a product called **FILELOK** that protects data as well as programs. It comes on fingerprinted disks that are blank except for the check routine. Any data you put on that disk can't be copied. This is a good solution for defense contractors, for example, that routinely deal with sensitive information. For added security, **FILELOK** comes with an option that encrypts the data to make it unreadable as well as uncopyable.

Vault was in the news about a year ago on account of a punitive protection scheme that became known as "Killer **PROLOK** .' Brotby says that Vault developed the infamous scheme at the request of its customers. Many of them had gotten tired of seeing their software pirated in huge quantities, especially overseas. In Mexico, for example, as many as 200 illegally made copies may exist for every legitimate piece of business software. In Singapore and Hong Kong, there are stores that openly sell cheap, pirated programs.

Killer **Prolok** , which Vault never planned to sell in the U.S., was going to teach pirates a lesson. If it couldn't find the laser holes on a bogus copy, it would wipe out data. If the disk were write-protected, Killer **Prolok** would wait and bash the next disk. Vault never released the product overseas either, but the very idea scared a lot of people.

The ADAPSO Key

Another physical software-protection technique is the ADAPSO key. ADAPSO, which is headquartered in Virginia and calls itself "the computer software and services industry association,' has invented an eleventh commandment: Thou shalt not dupe. However, injunction alone hasn't kept people from duplicating software, so ADAPSO has proposed a true hardwarebased protection scheme.

The ADAPSO system would require a lock box and a key. The key would be a small ROM chip that contained a small fraction of a protected program. The rest of the program would be on disk, but without the ROM key the program wouldn't work. The key would go into the lock box, a receptacle about the size of a pack of cigarettes, that, in turn, would plug into a computer's serial port. The lock box would include a pass-through to let you use the port for other things, but its main feature would be a set of slots, or keyholes.

Protected software would come with a disk and a key that you'd have

to plug into the lock box. The floppy portion of the program wouldn't be protected, so you could back it up all you wanted. ROM chips seldom go bad, so you wouldn't have to back them up. ROM chips are also damn hard to copy, so the vendor wouldn't have to worry about piracy.

So long as you left the key in the box, you could run the program any time. If you wanted to run it on another machine, you'd have to crawl behind your computer, pull out the key, and take it with you. You'd also have to have a serial port and a lock box on each machine. Finally, software vendors would have to set standards for the physical characteristics of the key and lock box for each program.

ROM chips are cheap, so the key would probably cost no more than $3 or $4. Lock boxes, even if they were made out of plastic in Taiwan, would cost more. The user, of course, would pay for this stuff.

In spite of ADAPSO's enthusiasm, its key and lockbox proposal has run into a very serious obstacle. Since ADAPSO is made up of competing software houses, the association has to get Justice Department permission before it can set standards. Otherwise the key might be seen as anticompetitive. Last December, ADAPSO asked for a formal "business review letter' from the Justice Department so it could proceed with the proposal. Nearly a year later, the department is still sitting on its hands. Says Dave Sturtevant, senior director of public communications for ADAPSO, "We're now on hold and about as frustrated as you can get.' He doesn't know why the Justice Department won't move.

A different hardware-based protection scheme that won't need a green light from the bureaucrats has been developed by Gordian Systems Inc. of Palo Alto, California. The Access Key system works by encrypting or scrambling a program. You can copy it all you want, but it's still scrambled. When you try to run the program, you get a logo screen, a series of flashing lights, and an invitation to type in a password.

The password is coded in the flashing lights, and once again you need a hardware "key' to figure it out. The key is a device about the size of a big eraser with a window at one end and a small, watch-sized liquid crystal panel on one side. When you hold the window up to the flashing lights, a battery-powered chip inside the key decodes the six-character password and flashes it on the liquid-crystal panel. You type it in at the keyboard, and the program will unscramble and run.

The Gordian Method

This scheme sounds like the same protection technique that is defeated by snatching the unscrambled version of the program out of RAM. There is, in fact, no defense against a utility that stops execution of a program and reads the contents of memory to disk. However, Gordian claims that the unscrambled version thus captured will not run. This defensive strategy works by taking a survey of the characteristics of the machine in which the program has been loaded at the time it is unscrambled. Thereafter, it checks every so often and if it finds it has been moved to another machine, it will refuse to run.

One slick feature of The Access Key system is that every time you run the program, it uses a **random** --number generator to produce a different six-character password. Thus, you never have to remember a password, but you can't run the program without a key. All keys have serial numbers so you can get a replacement if you're in a jam.

Just to make things harder for anyone to break the password code, The Access Key changes the coding scheme every 36 hours. When the protection software runs, it first checks the system date and modifies the encryption pattern accordingly. Only then does it throw up the logo screen and ask for the password. How does the key know what decoding method to use? It contains a quartz watch that tells it the date, and it, too, changes its decoding system every 36 hours.

The Gordian key is primarily an encryption /decryption device that is cunningly used to protect software. It can easily be used for data security. In this mode, you use a special software routine to make the key act as an encryptor. What started out as a readable file is now garbage. You use the same software routine and key to decrypt the file. This time, instead of keeping the unscrambled file in memory, the system writes it back to disk in readable form. Thus, the Gordian key is also a weapon in the battle for file security.

Protecting Your Data

There are many ways to keep data out of the wrong hands. One obvious

way is to keep it all on floppies and to keep the floppies locked up. If you need more storage, you can use removable hard disks and keep them in a safe.

If you have a fixed disk, one clever way to fool snoopers is to make it look as though the data isn't there. MagLock; by Flinder Software Laboratories, is a program that fiddles with the directory table so that DOS doesn't know your files are on the disk. MagLock can hide single files, subdirectories, or the contents of a whole disk, and it takes .85 seconds per file no matter how large or small. The data itself is untouched and will show up as bad sectors if you run a CHKDSK. Your files are still accessible to a snoop with a disk utility that reads individual sectors.

AST Research has developed a similar scheme called Knight Data Security Manager. It takes between 300K to 400K bytes on your hard disk and will ask for your name and password every time it boots. It is for multiuser systems, and it can be set up to keep certain people out of certain subdirectories. It also has an auditing option that records how much time people spend in which directories and how many times they try to get into off-limits directories. Knight is also set up so that if you boot from a floppy, DOS doesn't know the hard disk is there. If you ask for a directory, DOS gives you an "invalid drive spec' message. Knight also gives you the option of encrypting files.

Codes and Ciphers

A good way to keep data safe is to make it unreadable to anybody else. Soldiers and diplomats started doing this thousands of years ago when they invented codes, or ciphers.

Some codes are childishly simple. Take pig Latin. You code a word by moving the first consonant(s) to the end of the word and adding "ay.' Implesay odescay ancay ooklay ickytray. To decode, you do the reverse. This procedure for scrambling and unscrambling data is called an algorithm.

A simple computer-data encryption algorithm might increase the ASCII number of every character by a set number. If you choose to increase it by 20, the letter K (ASCII 75) would become a hyphen (ASCII 95), and the number 4 would become the letter H. To decode the file, you would reduce all the ASCII numbers by 20. The ASCII shift is the algorithm and the number 20 is the key, or password. An algorithm that consistently replaces one piece of data with another is called a substitution algorithm.

Permutation algorithms are another kind. If you were coding data in blocks of 64 characters, you might move every third character one place to the right. Even such a simple permutation algorithm, in combination with a substitution algorithm, would take a little while for an amateur to figure out. Real computer encryption algorithms, such as the U.S. government's Data Encryption Standard (DES) are, of course, much more complex (see "The Data Encryption Standard').

Although many encryption programs are on the market, one of the most widely available is included in SuperKey by Borland International. SuperKey actually contains two encryption methods, one a fullblown DES and the other a proprietary Borland algorithm that is not as complex but runs faster.

In both cases, the key or password can be up to 32 characters long. The encrypted file is written right over the original file, so no trace of the original is left on disk. If you type an encrypted file, all you get is smiling faces, Greek letters, and musical notes.

Borland also offers a text mode for encryption if you want to send code through a modem with a communications program that doesn't handle binary files. This mode produces a file that contains only the capital letters A through Z and is a good deal longer than the original file. If the person at the other end of the phone lines knows the password and also has a copy of SuperKey, he can decode the file.

Another company, United Software Security, produces an encrytion program called Privacy Plus. It, too, offers DES encryption as well as a simpler algorithm that is twice as fast. It also produces text mode for confidential data transmission. Like the Gordian system, it offers a hand-held device that decodes **randomly** generated passwords from flashing lights on the screen.

One interesting option on Privacy Plus is "master-keyed' encryptors, which permit multileveled, or "hierachical' security. These programs are set up so that no matter what key you use to encrypt a file, there is always another one--a sort of skeleton key--that will also decrypt it. This

makes sense in a company where a lot of people work with sensitive data. The security officer issues master-keyed versions of Privacy Plus to the employees. They can protect their data from each other and from outsiders, but if someone is out sick or forgets the password, the security officer can use the master key to decode anybody's files.

Hardware Encryption

Master keys may be handy for some purposes, but they point up an important weakness in encryption that's done with software: A clever hacker might fiddle with your program and insert a homemade master key, using it to decode everything you coded, and you might never know.

Encryption on a chip is the solution. Jones Futurex sells encryption boards for the IBM PC that incorporate tamperproof DES-standard chips made by Advanced Micro Devices and Western Digital.

Hardware encryption is also much faster than software encryption; Futurex boards clip along at 12,000 bytes per second. If you were running 1-2-3 or Word-Star, the board would automatically encrypt everything you wrote to disk and decrypt everything you read. Your application wouldn't run noticeably slower, your data would never be on-disk in readable form, and you wouldn't have to encrypt your files at the end of the day.

The more expensive Futurex boards go one step further, with an EPROM chip you can program with 64 encryption keys. Once again, a company's security officer would choose the on-board keys and also assign passwords to the system's users. The users' keys would work only with boards that were programmed to accept them. That way, disgruntled employees couldn't buy an identical Futurex board, take it home, and use it to decode company data. Even if they had valid user keys, the new board wouldn't be programmed with the matching on-board key.

As an extra barrier to sub-rosa decryption, 32 of the programmable keys are designed to disappear if you take the board out of the machine. That way you can't even sneak the company board home for a little decoding.

As a final precaution, Jones Futurex puts a steel case around the EPROM chip and fills the case with epoxy. There's a trip wire running through the epoxy, so if you try to cut through it, you'll destroy all the programmable encryption keys. This is real, hard-boiled security and probably more than most people need.

Know Your Enemy

However, the Futurex system underlines one of the differences between copy protection and data security. On the first battle front, it's a clear fight between copy protectors and copy busters. On the second, the combatants aren't always clear. Today's encryption algorithms are so arcane that anything that gets coded stays coded--unless there's an inside job.

But whichever side of the struggle you're on, the battle is far from over, and new weapons go into action every day.

SPECIAL FEATURES: illustration; table
DESCRIPTORS: Software Protection; Software; Copy Prevention Techniques; Software Piracy; Encryption; Data Security; Cryptography; Security
OPERATING PLATFORM: MSDOS
FILE SEGMENT: CD File 275

```
S1          17   (RANDOM)(2N)(SELECT? OR CHOOSE? OR DESIGNAT?)(2N)(ENCRYPTI-
              ON?()ALGORITHM?)
S2          11   S1 NOT PY>1996
S3          11   S2 NOT PD>19960801
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 340:CLAIMS(R)/US Patent 1950-03/Oct 30
         (c) 2003 IFI/CLAIMS(R)
File 342:Derwent Patents Citation Indx 1978-01/200345
         (c) 2003 Thomson Derwent
File 348:EUROPEAN PATENTS 1978-2003/Oct W03
         (c) 2003 European Patent Office
File 351:Derwent WPI 1963-2003/UD,UM &UP=200369
         (c) 2003  Thomson Derwent
File 654:US Pat.Full. 1976-2003/Oct 28
         (c) Format only 2003 The Dialog Corp.
```

3/9/1 **(Item 1 from file: 160)**
DIALOG(R)File 160:Gale Group PROMT(R)

01014238

The Integrator: Applying integrated systems in industry, engineering and commerce/New technical methods, litigation restrict illicit software use.

Mini Micro Systems April, 1984 p. 09-1141

Vault (Westlake, California) has developed a low-cost high quality software-protection diskette, Prolok, which places a minimal burden on end users, is applicable to different operating systems and requires no special hardware. The protection scheme is not vulnerable to a generalized software approach so that if someone manages to unlock 1 protected program the method would be of no use in unlocking other programs or additional protected copies of the same program. Prolok comes on a 'fingerprinted' diskette containing a variety of encryption and decryption algorithms. The fingerprint is unique to each disc with the fingerprints of different diskettes residing in different tracks and sectors. A variety of encryption and decryption algorithms are used, and Vault software uses **random** numbers to **select** an **encryption algorithm** . The protection also guards against use of debuggers or other software tools that could aid in copying protected software. The software detects exits from a decrypted program and erases the program from memory to prevent subsequent recovery. In addition, each byte of an encrypted program is dependent on all preceding encrypted bytes so that patching the program, even if possible, would lead to a wrong decryption. However, software-protection specialists say that even the combined features of Prolok will not be adequate as the difficulty of cracking a particular program will only serve as a challenge, the result being that commercially available programs for nullifying protection schemes will become prevalent. Vault maintains that it will always be at least 1 step ahead of those who create such unlocking programs.

COMPANY:
*Vault

PRODUCT: *Pkgd Software NEC (7372209)
EVENT: *Product Specifications (34)
COUNTRY: *United States (1USA)

```
Set     Items   Description
S1     271527   (ENCRYPT? OR ENCIPHER? OR CIPHER? OR CYPHER? ? OR ENCYPHER-
             ?)
S2      37509   S1(3N)(ALGORITHM? OR SYSTEM? OR FUNCTION? OR SCRAMBL? OR M-
            ATHEMATIC?()RULE?)
S3       1726   (MULTIPL? OR PLURAL? OR SEVERAL? OR MANY OR VARIOUS OR VAR-
            IET?)(5N)S2
S4     734800   RANDOM? OR PSUEDORANDOM? OR RNG? ?
S5        240   SIDE(N)SIDE
S6        117   (TELEPHON? OR PHONE?)()SCRAMBL?
S7    8587069   SELECT? OR ROTATE? OR CYCLE? OR SWITCH? OR ALTERNATING OR -
            CHOOSE? OR DESIGNAT?
S8        833   S3 AND (S4 OR S7)
S9       3508   (TELEPHON? OR TELECOM? OR PHON? OR MODEM)(5N)(S5 OR SCRAMB-
            L?)
S10       171   S3(S)(S4 OR S7)
S11       254   S9(S)S1
S12        77   S3(10N)(S4 OR S7)
S13        97   S9(3N)S1
S14         2   S11(S)(MULTIPL? OR PLURAL? OR SEVERAL? OR MANY OR VARIOUS -
              OR VARIET?)(S)S4
S15        57   S3(5N)(S4 OR S7)
S16        59   S14 OR S15
S17        34   RD (unique items)
S18        16   S17 NOT PY>1996
S19        11   S18 NOT PD=19960801:20010801
S20        11   S19 NOT PD=20010801:20031101
File 275:Gale Group Computer DB(TM) 1983-2003/Oct 30
         (c) 2003 The Gale Group
File  47:Gale Group Magazine DB(TM) 1959-2003/Oct 30
         (c) 2003 The Gale group
File  75:TGG Management Contents(R) 86-2003/Oct W2
         (c) 2003 The Gale Group
File 636:Gale Group Newsletter DB(TM) 1987-2003/Oct 30
         (c) 2003 The Gale Group
File  16:Gale Group PROMT(R) 1990-2003/Oct 30
         (c) 2003 The Gale Group
File 624:McGraw-Hill Publications 1985-2003/Oct 30
         (c) 2003 McGraw-Hill Co. Inc
File 484:Periodical Abs Plustext 1986-2003/Oct W3
         (c) 2003 ProQuest
File 813:PR Newswire 1987-1999/Apr 30
         (c) 1999 PR Newswire Association Inc
File 141:Readers Guide 1983-2003/Sep
         (c) 2003 The HW Wilson Co
File 239:Mathsci 1940-2003/Dec
         (c) 2003 American Mathematical Society
File 696:DIALOG Telecom. Newsletters 1995-2003/Oct 30
         (c) 2003 The Dialog Corp.
File 553:Wilson Bus. Abs. FullText 1982-2003/Sep
         (c) 2003 The HW Wilson Co
File 621:Gale Group New Prod.Annou.(R) 1985-2003/Oct 30
         (c) 2003 The Gale Group
File 674:Computer News Fulltext 1989-2003/Oct W4
         (c) 2003 IDG Communications
File  88:Gale Group Business A.R.T.S. 1976-2003/Oct 29
         (c) 2003 The Gale Group
File 369:New Scientist 1994-2003/Oct W4
         (c) 2003 Reed Business Information Ltd.
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 635:Business Dateline(R) 1985-2003/Oct 31
         (c) 2003 ProQuest Info&Learning
File  15:ABI/Inform(R) 1971-2003/Oct 31
         (c) 2003 ProQuest Info&Learning
File   9:Business & Industry(R) Jul/1994-2003/Oct 30
         (c) 2003 Resp. DB Svcs.
File  13:BAMP 2003/Oct W3
```

20/3,K/1 (Item 1 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01962169 SUPPLIER NUMBER: 18521173 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Fujitsu, Hitachi And NEC Developing Commerce System.
Newsbytes, pNEW07260012
July 26, 1996
LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 353 LINE COUNT: 00033

... certificates, a secure and reliable environment independent of specific Web browsers, high-level encryption and **selection** of any one of **multiple encryption algorithms** .
The credit card payment area of the new system also encompasses the Secure Electronic Transaction...

20/3,K/2 **(Item 2 from file: 275)**
DIALOG(R)File 275:Gale Group Computer DB(TM)

01900052 SUPPLIER NUMBER: 17957743 (USE FORMAT 7 OR 9 FOR FULL TEXT)
NetLock secures net transactions. (Hughes NetLock's NetLock cyptography software for encrypting packets on LANs and WANs) (Brief Article) (Product Announcement)
Pearlstein, Joanna
MacWEEK, v10, n5, p14(2)
Feb 5, 1996
DOCUMENT TYPE: Brief Article Product Announcement ISSN: 0892-8118
LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 236 LINE COUNT: 00023

... Macintosh versions of both the client and server are available. With NetLock Manager, administrators can **choose** from **several encryption algorithms** , including Data **Encryption** Standard (DES), Triple DES, RC2, RC4 and an encryption algorithm developed by Hughes. NetLock also...